



# **ATID Application Development Framework Reference Manual – Barcode Scanner**

Revision: Ver. 0.4

Date: January, 2014

ATID Co., Ltd

## Table of Contents

Table of Contents .....	2
Acronym .....	7
Revision History .....	8
1 .NET API Reference .....	9
1.1 Enumerations .....	9
1.1.1 BARCODE_RESULT .....	9
1.1.2 IMAGE_SIZE .....	10
1.1.3 IMAGE_FORMAT .....	10
1.1.4 SYMBOLOGIES_1D .....	10
1.1.5 SYMBOLOGIES_2DSWD .....	11
1.1.6 OCRDirection .....	12
1.1.7 OCRMode .....	12
1.2 Structures .....	13
1.2.1 BARCODECAPTUREPARAMS .....	13
1.2.2 GENERAL_CONFIG .....	13
1.2.3 SYMBOL_CONFIG .....	14
1.2.4 CONFIG_1D_CHINESE25 .....	14
1.2.5 CONFIG_1D_CODABAR .....	14
1.2.6 CONFIG_1D_CODE128 .....	15
1.2.7 CONFIG_1D_CODE39 .....	15
1.2.8 CONFIG_1D_CODE93 .....	16
1.2.9 CONFIG_1D_DISCRET25 .....	16
1.2.10 CONFIG_1D_INTERLEAVED25 .....	17
1.2.11 CONFIG_1D_ISBT128 .....	18
1.2.12 CONFIG_1D_MSI .....	18
1.2.13 CONFIG_1D_RSS .....	19
1.2.14 CONFIG_1D_UPCEAN .....	19
1.2.15 CONFIG_1D_CODE11 .....	22
1.2.16 CONFIG_2DSWD_AUSPOST .....	23
1.2.17 CONFIG_2DSWD_AZTEC .....	23
1.2.18 CONFIG_2DSWD_BPO .....	24
1.2.19 CONFIG_2DSWD_CANPOST .....	24
1.2.20 CONFIG_2DSWD_CHINAPOST .....	24
1.2.21 CONFIG_2DSWD_CODABAR .....	25
1.2.22 CONFIG_2DSWD_CODABLOCK .....	26
1.2.23 CONFIG_2DSWD_CODE11 .....	26
1.2.24 CONFIG_2DSWD_CODE128 .....	27

1.2.25	CONFIG_2DSWD_CODE16K.....	27
1.2.26	CONFIG_2DSWD_CODE32.....	27
1.2.27	CONFIG_2DSWD_CODE39.....	28
1.2.28	CONFIG_2DSWD_CODE49.....	29
1.2.29	CONFIG_2DSWD_CODE93.....	29
1.2.30	CONFIG_2DSWD_COMPOSITE .....	30
1.2.31	CONFIG_2DSWD_COUPONCODE .....	30
1.2.32	CONFIG_2DSWD_DATAMATRIX .....	31
1.2.33	CONFIG_2DSWD_DUTCHPOST .....	31
1.2.34	CONFIG_2DSWD_EAN13.....	31
1.2.35	CONFIG_2DSWD_ENA8 .....	32
1.2.36	CONFIG_2DSWD_GENERICCODE128 .....	33
1.2.37	CONFIG_2DSWD_GS1128.....	34
1.2.38	CONFIG_2DSWD_IATA25.....	34
1.2.39	CONFIG_2DSWD_INT25 .....	35
1.2.40	CONFIG_2DSWD_ISBT .....	35
1.2.41	CONFIG_2DSWD_JAPOST .....	36
1.2.42	CONFIG_2DSWD_KOREAPOST .....	36
1.2.43	CONFIG_2DSWD_MAXICODE .....	36
1.2.44	CONFIG_2DSWD_MAXICODE .....	37
1.2.45	CONFIG_2DSWD_MESA.....	37
1.2.46	CONFIG_2DSWD_MICROPDF.....	38
1.2.47	CONFIG_2DSWD_MSI.....	39
1.2.48	CONFIG_2DSWD_MX25.....	39
1.2.49	CONFIG_2DSWD_OCR .....	40
1.2.50	CONFIG_2DSWD_PDF417 .....	40
1.2.51	CONFIG_2DSWD_PLANET .....	41
1.2.52	CONFIG_2DSWD_PLESSEY .....	41
1.2.53	CONFIG_2DSWD_POSICODE.....	42
1.2.54	CONFIG_2DSWD_POSTNET .....	42
1.2.55	CONFIG_2DSWD_QR .....	43
1.2.56	CONFIG_2DSWD_RSS .....	43
1.2.57	CONFIG_2DSWD_STRT25.....	43
1.2.58	CONFIG_2DSWD_TELEPEN.....	44
1.2.59	CONFIG_2DSWD_TLC39 .....	44
1.2.60	CONFIG_2DSWD_TRIOPTIC .....	45
1.2.61	CONFIG_2DSWD_UPCA.....	45
1.2.62	CONFIG_2DSWD_UPCE.....	46
1.2.63	CONFIG_2DSWD_USPS4CB.....	47

1.2.64	CONFIG_2DSWD_IDTAG .....	48
1.3	Delegates .....	48
1.3.1	BARCODECALLBACK .....	48
1.4	Methods .....	49
1.4.1	Open .....	49
1.4.2	Close .....	49
1.4.3	IsOpened.....	49
1.4.4	SetCallback.....	49
1.4.5	GetVersionInfo .....	50
1.4.6	Start.....	50
1.4.7	StartScanRawData.....	51
1.4.8	Stop .....	51
1.4.9	StopScanRawData .....	51
1.4.10	GetScannedData.....	51
1.4.11	GetBarcodeRawData .....	52
1.4.12	InitCapture.....	53
1.4.13	DeinitCapture .....	53
1.4.14	SetPreviewHwnd .....	53
1.4.15	StartPreview .....	54
1.4.16	StopPreview .....	54
1.4.17	DoCapture .....	54
1.4.18	GetEnableStateAll.....	55
1.4.19	EnableSymbologiesAll.....	55
1.4.20	EnableSymbologies .....	56
1.4.21	DisableSymbologies.....	56
1.4.22	GetSymbologyConfig .....	57
1.4.23	SetSymbologyConfig .....	57
1.4.24	GetGeneralConfig .....	57
1.4.25	SetGeneralConfig .....	58
1.4.26	SetDefaultSymbol.....	58
1.4.27	SetScanningLightMode .....	59
1.4.28	SetLeaveLightsOn.....	59
1.4.29	GetLeaveLightsOn.....	60
1.5	Properties .....	60
1.5.1	IlluminationPowerLevel.....	60
2	C/C++ API Reference .....	61
2.1	Enumerations.....	61
2.1.1	BARCODE_RESULT.....	61
2.2	Constants.....	62

2.2.1	WM_BARCODE_ONSCANED .....	62
2.2.2	CAP_IMG_VGA .....	62
2.2.3	CAP_IMG_QVGA.....	62
2.2.4	CAP_IMG_PREVIEW .....	62
2.2.5	IMG_FORMAT_JPEG .....	62
2.2.6	IMG_FORMAT_BMP .....	62
2.3	Structures.....	63
2.3.1	BARCODECAPTUREPARAMS .....	63
2.3.2	BARCODE_GENERAL_CONFIG.....	63
2.3.3	BARCODE_SYMBOL_CONFIG .....	64
2.3.4	Symbol Configuration Structures .....	64
2.4	Callback function definition.....	64
2.4.1	BARCODECALLBACK .....	64
2.5	Methods.....	65
2.5.1	BarcodeOpen .....	65
2.5.2	BarcodeClose .....	65
2.5.3	BarcodeIsOpened.....	65
2.5.4	BarcodeSetCallback.....	66
2.5.5	BarcodeGetVersionInfo .....	66
2.5.6	BarcodeStart .....	66
2.5.7	BarcodeStartScanRawData .....	67
2.5.8	BarcodeStop .....	67
2.5.9	BarcodeStopScanRawData .....	67
2.5.10	BarcodeGetScannedData .....	67
2.5.11	BarcodeGetBarcodeRawData .....	68
2.5.12	BarcodeInitCapture.....	69
2.5.13	BarcodeDeinitCapture.....	69
2.5.14	BarcodeSetPreviewHwnd.....	69
2.5.15	BarcodeStartPreview .....	70
2.5.16	BarcodeStopPreview .....	70
2.5.17	BarcodeDoCapture .....	70
2.5.18	BarcodeGetEnableStateAll.....	71
2.5.19	BarcodeEnableSymbologiesAll .....	71
2.5.20	BarcodeEnableSymbologies .....	72
2.5.21	BarcodeDisableSymbologies.....	72
2.5.22	BarcodeGetSymbologyConfig .....	73
2.5.23	BarcodeSetSymbologyConfig .....	73
2.5.24	BarcodeGetGeneralConfig.....	74
2.5.25	BarcodeSetGeneralConfig .....	74

2.5.26	BarcodeSetDefaultSymbol.....	74
2.5.27	BarcodeSetScanningLightMode .....	75
2.5.28	BarcodeSetLeaveLightsOn.....	75
2.5.29	BarcodeGetLeaveLightsOn .....	76
2.5.30	BarcodeGetIlluminationPowerLevel .....	76
2.5.31	BarcodeSetIlluminationPowerLevel .....	76

## Acronym

modules	descriptions
<b>AADF</b>	ATID Application Development Framework

## Revision History

Version	Date	Reason	Description	Author
<b>0.1</b>	2012/01/17	Draft		Y. J. CHO
<b>0.2</b>	2013/01/25	Update	1. Symbology Configuration API 추가. 2. RAW Data 관련 내용 추가.	Y. J. CHO
<b>0.3</b>	2013/02/26	Update	1. LED Mode 설정 함수 추가.	Y. J. CHO
<b>0.4</b>	2014/01/10	Update	1. Illumination Power level 제어함수 추가.	Y. J. CHO



## 1 .NET API Reference

### 1.1 Enumerations

#### 1.1.1 BARCODE\_RESULT

함수의 호출 결과를 나타낸다.

- **BARCODE\_RESULT\_ALREADY\_BARCODE\_SELECTED**  
이미 바코드 Device가 선택되어 있음.
- **BARCODE\_RESULT\_ALREADY\_OPENED**  
바코드 Device가 이미 열려 있음.
- **BARCODE\_RESULT\_BARCODE\_INIT\_FAILURE**  
바코드 스캐너 초기화 실패.
- **BARCODE\_RESULT\_CAM\_SELECTED**  
CAM Device를 사용중임.
- **BARCODE\_RESULT\_FAILURE**  
기능 수행 실패.
- **BARCODE\_RESULT\_INVALID\_ARGS**  
유효하지 않은 parameter.
- **BARCODE\_RESULT\_INVALID\_DEVICE**  
바코드 스캐너가 장착되어 있지 않음.
- **BARCODE\_RESULT\_NOT\_OPENED**  
Open을 호출하지 않고 기능 수행을 시도 함.
- **BARCODE\_RESULT\_OUTOFMEMORY**  
바코드 스캐너 자원 할당 실패.
- **BARCODE\_RESULT\_SUCCESS**  
기능 수행 성공.
- **BARCODE\_RESULT\_UNSUPPORTED**  
지원 되지 않는 명령.

### 1.1.2 IMAGE\_SIZE

2D Barcode Scanner를 이용해서 캡처 할 이미지의 사이즈.

- **CAP\_IMG\_PREVIEW**

240 \* 180

- **CAP\_IMG\_QVGA**

320 \* 240

- **CAP\_IMG\_VGA**

640 \* 480

### 1.1.3 IMAGE\_FORMAT

2D Barcode Scanner를 이용해서 캡처 할 이미지의 포맷.

- **IMG\_FORMAT\_JPEG**

JPEG format

- **IMG\_FORMAT\_BMP**

BMP format

### 1.1.4 SYMBOLOGIES\_1D

1D 바코드 스캐너를 사용해서 디코딩 가능한 Symbology ID

- **NUM\_OF\_1D\_SYMBOLOGIES**

- **SYMBOL\_1D\_BOOKLAND**

- **SYMBOL\_1D\_CHINESE25**

- **SYMBOL\_1D\_CODABAR**

- **SYMBOL\_1D\_CODE11**

- **SYMBOL\_1D\_CODE128**

- **SYMBOL\_1D\_CODE39**

- **SYMBOL\_1D\_CODE93**

- **SYMBOL\_1D\_DISCRETE25**

- **SYMBOL\_1D\_EAN13**

- **SYMBOL\_1D\_EAN8**

- **SYMBOL\_1D\_INTERLEAVED25**

- **SYMBOL\_1D\_ISBT128**

- **SYMBOL\_1D\_MSI**

- **SYMBOL\_1D\_RSS14**

- **SYMBOL\_1D\_RSSEXPADED**

- **SYMBOL\_1D\_RSSLIMITED**

- SYMBOL\_1D\_TRIOPTIC39
- SYMBOL\_1D\_UCCEAN128
- SYMBOL\_1D\_UPCA
- SYMBOL\_1D\_UPCE
- SYMBOL\_1D\_UPCE1

#### 1.1.5 SYMBOLOGIES\_2DSWD

2D 바코드 스캐너를 사용해서 디코딩 가능한 Symbology ID

- NUM\_OF\_2DSWD\_SYMBOLOGIES
- SYMBOL\_2DSWD\_AUSPOST
- SYMBOL\_2DSWD\_AZTEC
- SYMBOL\_2DSWD\_BPO
- SYMBOL\_2DSWD\_CANPOST
- SYMBOL\_2DSWD\_CHINAPOST
- SYMBOL\_2DSWD\_CODABAR
- SYMBOL\_2DSWD\_CODABLOCK
- SYMBOL\_2DSWD\_CODE11
- SYMBOL\_2DSWD\_CODE128
- SYMBOL\_2DSWD\_CODE16K
- SYMBOL\_2DSWD\_CODE32
- SYMBOL\_2DSWD\_CODE39
- SYMBOL\_2DSWD\_CODE49
- SYMBOL\_2DSWD\_CODE93
- SYMBOL\_2DSWD\_COMPOSITE
- SYMBOL\_2DSWD\_COUPONCODE
- SYMBOL\_2DSWD\_DATAMATRIX
- SYMBOL\_2DSWD\_DUTCHPOST
- SYMBOL\_2DSWD\_EAN13
- SYMBOL\_2DSWD\_EAN8
- SYMBOL\_2DSWD\_GEN\_CODE128
- SYMBOL\_2DSWD\_GS1\_128
- SYMBOL\_2DSWD\_IATA25
- SYMBOL\_2DSWD\_IDTAG
- SYMBOL\_2DSWD\_INT25
- SYMBOL\_2DSWD\_ISBT
- SYMBOL\_2DSWD\_JAPOST
- SYMBOL\_2DSWD\_KOREAPOST

- SYMBOL\_2DSWD\_MATRIX25
- SYMBOL\_2DSWD\_MAXICODE
- SYMBOL\_2DSWD\_MESA
- SYMBOL\_2DSWD\_MICROPDF
- SYMBOL\_2DSWD\_MSI
- SYMBOL\_2DSWD\_OCR
- SYMBOL\_2DSWD\_PDF417
- SYMBOL\_2DSWD\_PLANET
- SYMBOL\_2DSWD\_PLESSEY
- SYMBOL\_2DSWD\_POSICODE
- SYMBOL\_2DSWD\_POSTNET
- SYMBOL\_2DSWD\_QR
- SYMBOL\_2DSWD\_RSS
- SYMBOL\_2DSWD\_STRT25
- SYMBOL\_2DSWD\_TELEPEN
- SYMBOL\_2DSWD\_TLCODE39
- SYMBOL\_2DSWD\_TRIOPTIC
- SYMBOL\_2DSWD\_UPCA
- SYMBOL\_2DSWD\_UPCE0
- SYMBOL\_2DSWD\_UPCE1
- SYMBOL\_2DSWD\_USPS4CB

#### **1.1.6 OCRDirection**

This enumeration is ignored and no longer supported.

- OCRDIRECTION\_BottomToTop
- OCRDIRECTION\_LeftToRight
- OCRDIRECTION\_RightToLeft
- OCRDIRECTION\_TopToBottom

#### **1.1.7 OCRMode**

This determines which OCR fonts (if any) are selected for decoding.

- OCRMODE\_A
- OCRMODE\_B
- OCRMODE\_DISABLED
- OCRMODE\_MICR\_UNSUPPORTED
- OCRMODE\_MONEY

## 1.2 Structures

### 1.2.1 BARCODECAPTUREPARAMS

2D Barcode Scanner를 이용해서 이미지를 캡처 할 때, 이미지 정보를 저장하는 구조체.

Public struct **BARCODECAPTUREPARAMS**

```
{
    Byte ImgFormat
    Byte ImgSize;
    String sFilePath;
};
```

- **ImgFormat**

이미지 포맷.

- **ImgSize**

이미지 사이즈.

- **sFilePath**

파일명을 포함한 이미지 저장 경로.

### 1.2.2 GENERAL\_CONFIG

1D 바코드 스캐너의 security level, scanning angle 상태를 읽거나 설정할 때 사용하는 구조체

Public struct **GENERAL\_CONFIG**

```
{
    bool bBiDirectionalRedundancy;
    int nAngle;
    int nLinearCodeSecurityLevel
}
```

- **bBiDirectionalRedundancy**

Returns whether a bar code must be successfully scanned in both direction(forward and reverse) before being decoded.

- **nAngle**

Scanning angle. 5=Narrow, 6=Wide

- **nLinearCodeSecurityLevel**

Four levels(1~4) of decode security for linear code types are supported. Select a higher security level for low levels of bar code quality. As security levels increase, the decoder's aggressiveness decrease.

### 1.2.3 SYMBOL\_CONFIG

Symbology 설정을 위하여 설정 정보를 담은 구조체.

Public struct **SYMBOL\_CONFIG**

```
{  
    IntPtr pConfigData;  
    Int Symbol;  
}
```

- **pConfigData**

설정을 변경할 Symbology의 구조체의 포인터

- **Symbol**

설정을 변경할 Symbology의 ID

### 1.2.4 CONFIG\_1D\_CHINESE25

1D바코드 스캐너에서 지원되는 Chinese 2 of 5 Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_1D\_CHINESE25**

```
{  
    bool bEnabled;  
}
```

- **bEnabled**

whether Chinese 2 of 5 is enabled or disabled

### 1.2.5 CONFIG\_1D\_CODABAR

1D바코드 스캐너에서 지원되는 Codabar Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_1D\_CODABAR**

```
{  
    bool bEnabled;  
    bool bCLSEditing;  
    bool bNOTISEditing;  
    short nMaxLength;  
    short nMinLength;  
}
```

- **bEnabled**

whether Codabar is enabled or disabled

- **bCLSIEditing**

whether CLSI Editing is enabled or disabled

- **bNOTISEditing**

whether NOTIS Editing is enabled or disabled

- **nMaxLength**

max length for Codabar

- **nMinLength**

min length for Codabar

### 1.2.6 CONFIG\_1D\_CODE128

1D바코드 스캐너에서 지원되는 Code 128 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_1D\\_CODE128](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether Code 128 is enabled or disabled

### 1.2.7 CONFIG\_1D\_CODE39

1D바코드 스캐너에서 지원되는 Code 39 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_1D\\_CODE39](#)

```
{
    bool bEnabled;
    bool bCheckDigitVerify;
    bool bCode32Prefix;
    bool bConvertCode39ToCode32;
    bool bEnableTriopticCode39;
    bool bFullAscii;
    bool bXmitCheckDigit;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Code 39 is enabled or disabled

- **bCheckDigitVerify**

Checks the integrity of a Code 39 symbol to ensure it complies with specified algorithms. Only Code 39 symbols which include a modulo 43 check digit are decoded.

- **bCode32Prefix**  
Appends the character 'A' to the start of the decode data if enabled.
- **bConvertCode39ToCode32**  
Converts Code 39 to Code 32.
- **bEnableTriopticCode39**  
whether Trioptic Code 39 is enabled or disabled.
- **bFullAscii**  
Enables or disables Code 39 Full ASCII
- **bXmitCheckDigit**  
whether Code 39 Check Digit is enabled or disabled.
- **nMaxLength**  
max length for Code 39
- **nMinLength**  
min length for Code 39

### 1.2.8 CONFIG\_1D\_CODE93

1D바코드 스캐너에서 지원되는 Code 93 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_1D\\_CODE93](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**  
whether Code 93 is enabled or disabled
- **nMaxLength**  
max length for Code 93
- **nMinLength**  
min length for Code 93

### 1.2.9 CONFIG\_1D\_DISCRET25

1D바코드 스캐너에서 지원되는 Discrete 2 of 5 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_1D\\_DISCRET25](#)



```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Discrete 2 of 5 is enabled or disabled

- **nMaxLength**

max length for Discrete 2 of 5

- **nMinLength**

min length for Discrete 2 of 5

### 1.2.10 CONFIG\_1D\_INTERLEAVED25

1D바코드 스캐너에서 지원되는 Interleaved 2 of 5 Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_1D\_DISCRET25**

```
{
    bool bEnabled;
    bool bConvertI2of5ToEAN13;
    bool bXmitCheckDigit;
    short nCheckDigitVerify;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Inverleaved 2 of 5 is enabled or disabled

- **bConvertI2of5ToEAN13**

Returns whether a 14-character I 2 of 5 code is converted to EAN-13 and transmitted to the host as EAN-13.

- **bXmitCheckDigit**

whether Interleaved 2 of 5 Check Digit is enabled or disabled.

- **nCheckDigitVerify**

Checks the integrity of an I 2 of 5 symbol to ensure it complies with specified algorithms, either USS (Uniform Symbology Specification), or OPCC (Optical Product Code Council).

- **nMaxLength**

max length for Inverleaved 2 of 5

- **nMinLength**

min length for Inverleaved 2 of 5

### 1.2.11 CONFIG\_1D\_ISBT128

1D바코드 스캐너에서 지원되는 ISBT 128 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_1D\\_ISBT128](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether ISBT 128 is enabled or disabled

### 1.2.12 CONFIG\_1D\_MSI

1D바코드 스캐너에서 지원되는 MSI Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_1D\\_MSI](#)

```
{
    bool bEnabled;
    bool bCheckDigitAlgorithm;
    bool bCheckDigitVerify;
    bool bXmitCheckDigit;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether MSI is enabled or disabled

- **bCheckDigitAlgorithm**

When Two MSI check digits is selected, sets an additional verification is required to ensure integrity. Two following algorithms may be selected: MOD10/MOD11 or MOD10/MOD10.

- **bCheckDigitVerify**

Sets the number of check digits at the end of the bar code that verify the integrity of the data. At least one check digit is always required. Check digits are not automatically transmitted with the data.

- **bXmitCheckDigit**

whether MSI Check Digit is transmitted with the data.

- **nMaxLength**

max length for MSI

- **nMinLength**  
min length for MSI

### 1.2.13 CONFIG\_1D\_RSS

1D바코드 스캐너에서 지원되는 RSS Symbology의 설정 정보를 저장하는 구조체

```
Public struct CONFIG_1D_MSI
{
    bool bRSS14Enabled;
    bool bRSSExpandedEnabled;
    bool bRSSLimitedEnabled;
    bool bConvertRSSToUPCEAN;
}
```

- **bRSS14Enabled**  
whether RSS 14 is enabled or disabled
- **bRSSExpandedEnabled**  
whether RSS Expanded is enabled or disabled
- **bRSSLimitedEnabled**  
whether RSS Limited is enabled or disabled
- **bConvertRSSToUPCEAN**  
convert to RSS to UPCEAN

### 1.2.14 CONFIG\_1D\_UPCEAN

1D바코드 스캐너에서 지원되는 UPC/EAN Symbology의 설정 정보를 저장하는 구조체

```
Public struct CONFIG_1D_UPCEAN
{
    bool bBooklandEnabled;
    bool bConvertEAN8ToEAN13;
    bool bConvertUPCE1ToUPCA;
    bool bConvertUPCEToUPCA;
    bool bEAN13Enabled;
    bool bEAN8Enabled;
    bool bEANZeroExtend;
    bool bUCCCouponExtendedCode;
    bool bUCCEAN128Enabled;
    bool bUPCAEnabled;
    bool bUPCE1Enabled;
}
```

```

bool bUPCEEnabled;
bool bXmitUPCACheckChar;
bool bXmitUPCE1CheckChar;
bool bXmitUPCECheckChar;
short nDecodeUPCEANSupplemental;
short nDecodeUPCEANSupplementalRedundancy;
short nUPCAPreamble;
short nUPCE1Preamble;
short nUPCEANSecurityLevel;
short nUPCEPreamble;
}

```

- **bBooklandEnabled**

whether Bookland is enabled or disabled

- **bConvertEAN8ToEAN13**

When EAN Zero Extend is enabled, labels the extended symbol as either an EAN-13 bar code, or an EAN-8 bar code.

When EAN Zero Extend is disabled, this parameter has no effect on bar code data.

- **bConvertUPCE1ToUPCA**

Converts UPC-E1 (zero suppressed) decoded data to UPC-A format before transmission. After conversion, data follows UPC-A format and is affected by UPC-A programming selections (e.g., Preamble, Check Digit).

- **bConvertUPCEToUPCA**

Converts UPC-E (zero suppressed) decoded data to UPC-A format before transmission. After conversion, data follows UPC-A format and is affected by UPC-A programming selections (e.g., Preamble, Check Digit).

- **bEAN13Enabled**

whether EAN-13 is enabled or disabled

- **bEAN8Enabled**

whether EAN-8 is enabled or disabled

- **bEANZeroExtend**

When enabled, five leading zeros are added to decoded EAN-8 symbols to make them compatible in format to EAN-13 symbols.

- **bUCCCouponExtendedCode**

whether UCC Coupon is enabled or disabled

- **bUCCEAN128Enabled**

whether UCCEAN 128 is enabled or disabled

- **bUPCAEnabled**

whether UPC-A is enabled or disabled

- **bUPCE1Enabled**

whether UPC-E1 is enabled or disabled

- **bUPCEEnabled**

whether UPC-E is enabled or disabled

- **bXmitUPCACheckChar**

Transmits the symbol with or without the UPC-A check digit.

- **bXmitUPCE1CheckChar**

Transmits the symbol with or without the UPC-E1 check digit.

- **bXmitUPCECheckChar**

Transmits the symbol with or without the UPC-E check digit.

- **nDecodeUPCEANSupplemental**

Sets Decode UPC/EAN Supplementals option. Supplementals are additionally appended characters (2 or 5) according to specific code format conventions (e.g., UPC A+2, UPC E+2, EAN 8+2). Three options are available.

0 : Decode supplementals

1 : Ignore supplementals

2 : Auto discriminate supplementals

- **nDecodeUPCEANSupplementalRedundancy**

When Autodiscriminate UPC/EAN Supplementals selected, adjusts the number of times a symbol without supplementals is decoded before transmission. The range is from 2 to 20 times. Five or above is recommended when decoding a mix of UPC/EAN symbols with and without supplementals, and the autodiscriminate option is selected.

this is an integer value in the range.[2..20]

- **nUPCAPreamble**

Returns the selected UPC-A Preamble option: transmit system character only, transmit system character and country code ("0" for USA), or no preamble transmitted. The lead-in characters are considered part of the symbol.

0 : No preamble

1 : System character

2 : System character, country code

- **nUPCE1Preamble**

Returns the selected UPC-E1 Preamble option: transmit system character only, transmit system character and country code ("0" for USA), or no preamble transmitted. The lead-in characters are considered part of the symbol.

0 : No preamble

1 : System character

2 : System character, country code

- **nUPCEANSecurityLevel**

Sets the UPC/EAN Security Level. There are four levels of decode security for UPC/EAN bar codes. Select a higher level of security are provided for decreasing levels of bar code quality. There is an inverse relationship between security and decoder aggressiveness, so be sure to choose only that level of security necessary for any given application.

- UPC/EAN Security Level 0: This default setting allows the decoder to operate in its most aggressive state, while providing sufficient security in decoding "in-spec" UPC/EAN bar codes.

- UPC/EAN Security Level 1: As bar code quality levels diminish, certain characters become prone to misdecodes before others (i.e., 1, 2, 7, 8). If you are experiencing misdecodes of poorly printed bar codes, and the misdecodes are limited to these characters, select this security level.

- UPC/EAN Security Level 2: If you are experiencing misdecodes of poorly printed bar codes, and the misdecodes are not limited to characters 1, 2, 7, and 8, select this security level.

- UPCI EAN Security Level 3: If you have tried Security Level 2, and are still experiencing misdecodes, select this security level. Be advised, selecting this option is an extreme measure against mis-decoding severely out of spec bar codes. Selection of this level of security significantly impairs the decoding ability of the decoder. If this level of security is necessary you should try to improve the quality of your bar codes.

- **nUPCEPreamble**

Sets the UPC-E Preamble option. Three options are given for lead-in characters for UPC-E symbols transmitted to the host device: transmit system character only, transmit system character and country code ("0" for USA), and no preamble transmitted. The lead-in characters are considered part of the symbol.

0 : No preamble

1 : System character

2 : System character, country code

### 1.2.15 CONFIG\_1D\_CODE11

1D바코드 스캐너에서 지원되는 Code 11 Symbolology의 설정 정보를 저장하는 구조체

```
Public struct CONFIG_1D_CODE11
{
    bool bEnabled;
    bool bCheckDigitVerify;
    bool bXmitCheckDigit;
```

```

short nMaxLength;
short nMinLength;
}

```

- **bEnabled**

whether Code 11 is enabled or disabled

- **bCheckDigitVerify**

Sets the number of check digits at the end of the bar code that verify the integrity of the data. At least one check digit is always required. Check digits are not automatically transmitted with the data.

- **bXmitCheckDigit**

whether Code 11 Check Digit is transmitted with the data.

- **nMaxLength**

max length for Code 11

- **nMinLength**

min length for Code 11

#### 1.2.16 CONFIG\_2DSWD\_AUSPOST

2D바코드 스캐너에서 지원되는 Australian Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_AUSPOST](#)

```

{
    bool bEnabled;
}

```

- **bEnabled**

whether Australian Postal Code is enabled or disabled

#### 1.2.17 CONFIG\_2DSWD\_AZTEC

2D바코드 스캐너에서 지원되는 Aztec Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_AUSPOST](#)

```

{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}

```

- **bEnabled**

whether AZTEC is enabled or disabled

- **nMaxLength**  
max length for AZTEC
- **nMinLength**  
min length for AZTEC

#### 1.2.18 CONFIG\_2DSWD\_BPO

2D바코드 스캐너에서 지원되는 British Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_BPO](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**  
whether British Post is enabled or disabled

#### 1.2.19 CONFIG\_2DSWD\_CANPOST

2D바코드 스캐너에서 지원되는 Canadian Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CANPOST](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**  
whether Canadian Post is enabled or disabled

#### 1.2.20 CONFIG\_2DSWD\_CHINAPOST

2D바코드 스캐너에서 지원되는 Chinese Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CHINAPOST](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**



whether China post is enabled or disabled

- **nMaxLength**

max length for China post

- **nMinLength**

min length for China post

### 1.2.21 CONFIG\_2DSWD\_CODABAR

2D바코드 스캐너에서 지원되는 Codabar Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODABAR](#)

```
{
    bool bEnabled;
    bool bCheckCharOn;
    bool bSSXmit;
    bool bXmitCheckChar;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

- **whether Codabar is enabled or disabled**

- **bCheckCharOn**

BOOL variable that determines if the engine will read Codabar bar codes with or without check characters. If TRUE, the engine only decodes Codabar codes with a check character. If FALSE, the decoder decodes codes with or without a check character.

- **bSSXmit**

BOOL variable that determines if the start and stop characters are returned in the data string after a successful Codabar decode. If bSSXmit is TRUE, the start and stop characters are included. If FALSE, they are not included.

- **bXmitCheckChar**

BOOL variable that determines if the engine will return the check character as part of the data string after a successful decode. If TRUE, the engine returns the check character. If FALSE the check character is not returned.

- **nMaxLength**

max length for Codabar

- **nMinLength**

min length for Codabar

### 1.2.22 CONFIG\_2DSWD\_CODABLOCK

2D바코드 스캐너에서 지원되는 Codablock F Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODABLOCK](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Codablock F is enabled or disabled

- **nMaxLength**

max length for Codablock F

- **nMinLength**

min length for Codablock F

### 1.2.23 CONFIG\_2DSWD\_CODE11

2D바코드 스캐너에서 지원되는 Code 11 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE11](#)

```
{
    bool bEnabled;
    bool bTwoCheckDigits;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Codablock F is enabled or disabled

- **bTwoCheckDigits**

BOOL variable that determines if the engine is decoding Code 11 bar codes that have two check digits. If TRUE, the engine is decoding Code 11 bar codes that have two check digits. If FALSE the engine decodes Code 11 bar codes as if they were printed with only one check digit.

- **nMaxLength**

max length for Code 11

- **nMinLength**

min length for Code 11

#### 1.2.24 CONFIG\_2DSWD\_CODE128

2D바코드 스캐너에서 지원되는 Code 128 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE128](#)

```
{  
    bool bEnabled;  
    short nMaxLength;  
    short nMinLength;  
}
```

- **bEnabled**

whether Code 128 is enabled or disabled

- **nMaxLength**

max length for Code 128

- **nMinLength**

min length for Code 128

#### 1.2.25 CONFIG\_2DSWD\_CODE16K

2D바코드 스캐너에서 지원되는 Code 16K Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE16K](#)

```
{  
    bool bEnabled;  
    short nMaxLength;  
    short nMinLength;  
}
```

- **bEnabled**

whether Code 16K is enabled or disabled

- **nMaxLength**

max length for Code 16K

- **nMinLength**

min length for Code 16K

#### 1.2.26 CONFIG\_2DSWD\_CODE32

2D바코드 스캐너에서 지원되는 Code 32 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE32](#)

```
{
```

```
bool bEnabled;
}
```

- **bEnabled**  
whether Code 16K is enabled or disabled

### 1.2.27 CONFIG\_2DSWD\_CODE39

2D바코드 스캐너에서 지원되는 Code 39 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE39](#)

```
{
    bool bEnabled;
    bool bAppend;
    bool bCheckCharOn;
    bool bFullAscii;
    bool bSSXmit;
    bool bXmitCheckChar;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**  
whether Code 39 is enabled or disabled

- **bAppend**

BOOL variable that determines if the engine should append together and buffer up Code 39 symbols that start with a space (excluding the start and stop characters). The engine stores the symbols in the order in which they are read. It returns the data after a Code 39 symbol with no leading space is read. The return data has the leading spaces removed. If TRUE, the append feature is enabled. If FALSE, the append feature is disabled.

- **bCheckCharOn**

BOOL variable that determines if the engine will read Code 39 bar codes with or without check characters. If TRUE, the engine only decodes Code 39 codes with a check character. If FALSE, the decoder decodes codes with or without a check character.

- **bFullAscii**

BOOL variable that determines if certain character pairs within the bar code symbol are interpreted and returned as a single character. If bFullAscii is TRUE, interpretation is enabled. If FALSE, no interpretation is attempted.

- **bSSXmit**

BOOL variable that determines if the start and stop characters are returned in the data string after a successful Code 39 decode. If bSSXmit is TRUE, the start and stop characters are included. If FALSE, they are not included.

- **bXmitCheckChar**

BOOL variable that determines if the engine will return the check character as part of the data string after a successful decode. If TRUE, the engine returns the check character. If FALSE, the check character is not returned.

- **nMaxLength**

max length for Code 39

- **nMinLength**

min length for Code 39

### 1.2.28 CONFIG\_2DSWD\_CODE49

2D바코드 스캐너에서 지원되는 Code 49 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE49](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Code 49 is enabled or disabled

- **nMaxLength**

max length for Code 49

- **nMinLength**

min length for Code 49

### 1.2.29 CONFIG\_2DSWD\_CODE93

2D바코드 스캐너에서 지원되는 Code 93 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_CODE93](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Code 93 is enabled or disabled

- **nMaxLength**

max length for Code 93

- **nMinLength**

min length for Code 93

### 1.2.30 CONFIG\_2DSWD\_COMPOSITE

2D바코드 스캐너에서 지원되는 Composite Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_COMPOSITE](#)

```
{
    bool bEnabled;
    bool bCompositeOnUpcEan;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Composite is enabled or disabled

- **bCompositeOnUpcEan**

BOOL variable that contains the enabled state of EAN•UCC Composite code associated with EAN and UPC codes.

- **nMaxLength**

max length for Composite

- **nMinLength**

min length for Composite

### 1.2.31 CONFIG\_2DSWD\_COUPONCODE

2D바코드 스캐너에서 지원되는 Coupon Code Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_COUPONCODE](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether Coupon code is enabled or disabled

### 1.2.32 CONFIG\_2DSWD\_DATAMATRIX

2D바코드 스캐너에서 지원되는 Data Matrix Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_DATAMATRIX](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Data Matrix is enabled or disabled

- **nMaxLength**

max length for Data Matrix

- **nMinLength**

min length for Data Matrix

### 1.2.33 CONFIG\_2DSWD\_DUTCHPOST

2D바코드 스캐너에서 지원되는 KIX(Netherlands) Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_DUTCHPOST](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether KIX(Netherlands) Post is enabled or disabled

### 1.2.34 CONFIG\_2DSWD\_EAN13

2D바코드 스캐너에서 지원되는 Ean-13 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_EAN13](#)

```
{
    bool bEnabled;
    bool bAddenda2Digit;
    bool bAddenda5Digit;
    bool bAddendaReq;
}
```

```
bool bAddendaSeparator;
bool bXmitCheckChar;
}
```

- **bEnabled**

whether EAN 13 is enabled or disabled

- **bAddenda2Digit**

BOOL variable that determines if the engine will look for a 2 digit addenda at the end of the EAN/JAN-13 barcode. If TRUE, and an addenda is present, the engine adds the two digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddenda5Digit**

BOOL variable that determines if the engine will look for a 5 digit addenda at the end of the EAN/JAN-13 barcode. If TRUE, and an addenda is present, the engine adds the five digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddendaReq**

BOOL variable that determines if the engine will decode only EAN/JAN-13 bar codes that have a 2 or 5 digit addenda. If TRUE, the engine decodes only EAN symbols with an addenda. If FALSE, the engine decodes all enabled EAN/JAN-13 symbols.

- **bAddendaSeparator**

BOOL variable that determines if there is a space character between the data from the bar code and the data from the addenda. If TRUE, there is a space. If FALSE, there is no space.

- **bXmitCheckChar**

BOOL variable that determines if the engine will return the check character as part of the data string after a successful decode. If TRUE, the engine returns the check character. If FALSE, the check character is not returned.

### 1.2.35 CONFIG\_2DSWD\_EAN8

2D바코드 스캐너에서 지원되는 Ean-8 Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_2DSWD\_EAN8**

```
{
    bool bEnabled;
    bool bAddenda2Digit;
    bool bAddenda5Digit;
    bool bAddendaReq;
    bool bAddendaSeparator;
```



```
bool bXmitCheckChar;
}
```

- **bEnabled**

whether EAN 13 is enabled or disabled

- **bAddenda2Digit**

BOOL variable that determines if the engine will look for a 2 digit addenda at the end of the EAN/JAN-8 barcode. If TRUE, and an addenda is present, the engine adds the two digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddenda5Digit**

BOOL variable that determines if the engine will look for a 5 digit addenda at the end of the EAN/JAN-8 barcode. If TRUE, and an addenda is present, the engine adds the five digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddendaReq**

BOOL variable that determines if the engine will decode only EAN/JAN-8 bar codes that have a 2 or 5 digit addenda. If TRUE, the engine decodes only EAN/JAN-8 symbols with an addenda. If FALSE, the engine decodes all enabled EAN/JAN-8 symbols.

- **bAddendaSeparator**

BOOL variable that determines if there is a space character between the data from the bar code and the data from the addenda. If TRUE, there is a space. If FALSE, there is no space.

- **bXmitCheckChar**

BOOL variable that determines if the engine will return the check character as part of the data string after a successful decode. If TRUE, the engine returns the check character. If FALSE, the check character is not returned.

### 1.2.36 CONFIG\_2DSWD\_GENERICCODE128

2D바코드 스캐너에서 지원되는 Generic Code 128 Symbology의 설정 정보를 저장하는 구조체

```
Public struct CONFIG_2DSWD_GENERICCODE128
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**  
whether Generic Code 128 is enabled or disabled
- **nMaxLength**  
max length for Generic Code 128
- **nMinLength**  
min length for Generic Code 128

### 1.2.37 CONFIG\_2DSWD\_GS1128

2D바코드 스캐너에서 지원되는 GS1 128 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_GS1128](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**  
whether GS1 128 is enabled or disabled
- **nMaxLength**  
max length for GS1 128
- **nMinLength**  
min length for GS1 128

### 1.2.38 CONFIG\_2DSWD\_IATA25

2D바코드 스캐너에서 지원되는 IATA 2 of 5 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_IATA25](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**  
whether Straight 2 of 5 IATA is enabled or disabled
- **nMaxLength**  
max length for Straight 2 of 5 IATA
- **nMinLength**

min length for Straight 2 of 5 IATA

### 1.2.39 CONFIG\_2DSWD\_INT25

2D바코드 스캐너에서 지원되는 Interleaved 2 of 5 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_INT25](#)

```
{
    bool bEnabled;
    bool bCheckDigitOn;
    bool bXmitCheckDigit;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Straight 2 of 5 IATA is enabled or disabled

- **bCheckDigitOn**

BOOL variable that determines if the engine will read Interleaved 2 of 5 bar codes with or without check characters. If TRUE, the engine only decodes Interleaved 2 of 5 codes with a check digit. If FALSE, the decoder decodes codes with or without a check digit.

- **bXmitCheckDigit**

BOOL variable that determines if the engine will return the check digit as part of the data string after a successful decode. If TRUE, the engine returns the check digit. If FALSE, the check digit is not returned.

- **nMaxLength**

max length for Straight 2 of 5 IATA

- **nMinLength**

min length for Straight 2 of 5 IATA

### 1.2.40 CONFIG\_2DSWD\_ISBT

2D바코드 스캐너에서 지원되는 ISBT Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_ISBT](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether ISBT 128 is enabled or disabled

#### 1.2.41 CONFIG\_2DSWD\_JAPOST

2D바코드 스캐너에서 지원되는 Japanese Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_JAPOST](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether Japanese Post is enabled or disabled

#### 1.2.42 CONFIG\_2DSWD\_KOREAPOST

2D바코드 스캐너에서 지원되는 Korean Post Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_KOREAPOST](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Korean Post is enabled or disabled

- **nMaxLength**

max length for Korean Post

- **nMinLength**

min length for Korean Post

#### 1.2.43 CONFIG\_2DSWD\_MAXICODE

2D바코드 스캐너에서 지원되는 Maxi Code Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_MAXICODE](#)

```
{
    bool bEnabled;
    bool bCarrierMsgOnly;
}
```

```

short nMaxLength;
short nMinLength;
}
- bEnabled
  whether Maxi Code is enabled or disabled
- bCarrierMsgOnly
  This parameter is ignored and no longer supported.
- nMaxLength
  max length for Maxi Code
- nMinLength
  min length for Maxi Code

```

#### 1.2.44 CONFIG\_2DSWD\_MAXICODE

2D바코드 스캐너에서 지원되는 Maxi Code Symbology의 설정 정보를 저장하는 구조체

```

Public struct CONFIG_2DSWD_MAXICODE
{
  bool bEnabled;
  bool bCarrierMsgOnly;
  short nMaxLength;
  short nMinLength;
}
- bEnabled
  whether Maxi Code is enabled or disabled
- bCarrierMsgOnly
  This parameter is ignored and no longer supported.
- nMaxLength
  max length for Maxi Code
- nMinLength
  min length for Maxi Code

```

#### 1.2.45 CONFIG\_2DSWD\_MESA

2D바코드 스캐너에서 지원되는 Mesa Symbology의 설정 정보를 저장하는 구조체

```

Public struct CONFIG_2DSWD_MESA
{
  bool b1MSEnabled;

```

```

bool b3MSEnabled;
bool b9MSEnabled;
bool bEMSEnabled;
bool bIMSEnabled;
bool bUMSEnabled;
}

- b1MSEnabled
  BOOL variable that contains the enabled state of Code 128 Mesa.
  TRUE = Enabled, FALSE = Disabled.

- b3MSEnabled
  BOOL variable that contains the enabled state of Code 39 Mesa.
  TRUE = Enabled, FALSE = Disabled.

- b9MSEnabled
  BOOL variable that contains the enabled state of Code 93 Mesa.

- bEMSEnabled
  BOOL variable that contains the enabled state of EAN13 Mesa.
  TRUE = Enabled, FALSE = Disabled.

- bIMSEnabled
  BOOL variable that contains the enabled state of Interleaved 2 of 5 Mesa.
  TRUE = Enabled, FALSE = Disabled.

- bUMSEnabled
  BOOL variable that contains the enabled state of UPCA Mesa.
  TRUE = Enabled, FALSE = Disabled.

```

#### 1.2.46 CONFIG\_2DSWD\_MICROPDF

2D바코드 스캐너에서 지원되는 Micro PDF의 설정 정보를 저장하는 구조체

```

Public struct CONFIG_2DSWD_MICROPDF
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}

- bEnabled
  whether Micro PDF is enabled or disabled

- nMaxLength
  max length for Micro PDF

- nMinLength

```

min length for Micro PDF

#### 1.2.47 CONFIG\_2DSWD\_MSI

2D바코드 스캐너에서 지원되는 MSI Code Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_MSI](#)

```
{
    bool bEnabled;
    bool bXmitCheckChar;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Maxi Code is enabled or disabled

- **bXmitCheckChar**

BOOL variable that determines if the engine will return the check character as part of the data string after a successful decode. If TRUE, the engine returns the check character. If FALSE, the check character is not returned.

- **nMaxLength**

max length for MSI

- **nMinLength**

min length for MSI

#### 1.2.48 CONFIG\_2DSWD\_MX25

2D바코드 스캐너에서 지원되는 Micro PDF의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_MX25](#)

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Matrix 2 of 5 is enabled or disabled

- **nMaxLength**

max length for Matrix 2 of 5

- **nMinLength**

min length for Matrix 2 of 5

#### 1.2.49 CONFIG\_2DSWD\_OCR

2D바코드 스캐너에서 지원되는 OCR Symbology의 설정 정보를 저장하는 구조체

Public struct `CONFIG_2DSWD_OCR`

```
{
    String CheckChar;
    String GroupG;
    String GroupH;
    String Template;
    OCRMode nFont;
}
```

- **CheckChar**

A null-terminated string that represents a check character position in the template strings.

- **GroupG**

A null-terminated string that represents a list of characters that can be substituted for the lower-case 'g' in the template strings.

- **GroupH**

A null-terminated string that represents a list of characters that can be substituted for the lower-case 'h' in the template strings.

- **Template**

A null-terminated string that indicates one or more template patterns for the OCR decode. The following characters are allowed:

A-Z : capital letters are matched as is

d : a digit from 0 - 9

a : alphanumeric character

l : alphabetic letter

g : any character specified in group G

h : any character specified in group H

#### 1.2.50 CONFIG\_2DSWD\_PDF417

2D바코드 스캐너에서 지원되는 PDF417 Symbology의 설정 정보를 저장하는 구조체

Public struct `CONFIG_2DSWD_PDF417`

```
{
    bool bEnabled;
    short nMaxLength;
    short nMinLength;
```



- ```
}
- bEnabled
  whether PDF417 is enabled or disabled
- nMaxLength
  max length for PDF417
- nMinLength
  min length for PDF417
```

### 1.2.51 CONFIG\_2DSWD\_PLANET

2D바코드 스캐너에서 지원되는 Planet Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_PLANET](#)

```
{
  bool bEnabled;
  bool bXmitCheckDigit;
}
```

- **bEnabled**  
whether Planet is enabled or disabled
- **bXmitCheckDigit**  
BOOL variable that determines if the engine will return the check digit as part of the data string after a successful decode. If TRUE, the engine returns the check digit. If FALSE, the check digit is not returned.

### 1.2.52 CONFIG\_2DSWD\_PLESSEY

2D바코드 스캐너에서 지원되는 Plessey Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_PLESSEY](#)

```
{
  bool bEnabled;
  short nMaxLength;
  short nMinLength;
}
```

- **bEnabled**  
whether Plessey Code is enabled or disabled
- **nMaxLength**  
max length for Plessey Code
- **nMinLength**  
min length for Plessey Code

### 1.2.53 CONFIG\_2DSWD\_POSICODE

2D바코드 스캐너에서 지원되는 Posi Code Symbology의 설정 정보를 저장하는 구조체

Public struct `CONFIG_2DSWD_POSICODE`

```
{
    bool bEnabled;
    short wLimited;
    short nMaxLength;
    short nMinLength;
}
```

- **bEnabled**

whether Posi Code is enabled or disabled

- **wLimited**

short variable that reflects if Posicode Limited A or Posicode Limited B decoding is enabled. A value of 1 indicates Posicode Limited A is enabled, and a value of 2 indicates Posicode Limited B decoding is enabled. A value of 0 indicates that decoding of both Limited A and Limited B is disabled.

- **nMaxLength**

max length for Posi Code

- **nMinLength**

min length for Posi Code

### 1.2.54 CONFIG\_2DSWD\_POSTNET

2D바코드 스캐너에서 지원되는 Postnet Symbology의 설정 정보를 저장하는 구조체

Public struct `CONFIG_2DSWD_POSTNET`

```
{
    bool bEnabled;
    bool bXmitCheckDigit;
}
```

- **bEnabled**

whether Postnet is enabled or disabled

- **bXmitCheckDigit**

BOOL variable that determines if the engine will return the check digit as part of the data string after a successful decode. If TRUE, the engine returns the check digit. If FALSE, the check digit is not returned.

### 1.2.55 CONFIG\_2DSWD\_QR

2D바코드 스캐너에서 지원되는 QR Code Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_QR](#)

```
{  
    bool bEnabled;  
    short nMaxLength;  
    short nMinLength;  
}
```

- **bEnabled**

whether QR Code is enabled or disabled

- **nMaxLength**

max length for QR Code

- **nMinLength**

min length for QR Code

### 1.2.56 CONFIG\_2DSWD\_RSS

2D바코드 스캐너에서 지원되는 RSS Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_RSS](#)

```
{  
    bool bEnabled;  
    short nMaxLength;  
    short nMinLength;  
}
```

- **bEnabled**

whether RSS is enabled or disabled

- **nMaxLength**

max length for RSS

- **nMinLength**

min length for RSS

### 1.2.57 CONFIG\_2DSWD\_STRT25

2D바코드 스캐너에서 지원되는 Straight 2 of 5 Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_STRT25](#)

```
{
```

```

bool bEnabled;
short nMaxLength;
short nMinLength;
}
- bEnabled
  whether Straight 2 of 5 is enabled or disabled
- nMaxLength
  max length for Straight 2 of 5
- nMinLength
  min length for Straight 2 of 5

```

### 1.2.58 CONFIG\_2DSWD\_TELEPEN

2D바코드 스캐너에서 지원되는 Telepen Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_2DSWD\_STRT25**

```

{
  bool bEnabled;
  bool bOldStyle;
  short nMaxLength;
  short nMinLength;
}

```

- **bEnabled**  
whether Telepen is enabled or disabled
- **bOldStyle**  
BOOL variable that reflects if the engine is configured to reads Telepen labels that were encoded with either the original or the AIM specification.
- **nMaxLength**  
max length for Telepen
- **nMinLength**  
min length for Telepen

### 1.2.59 CONFIG\_2DSWD\_TLC39

2D바코드 스캐너에서 지원되는 TLC39 Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_2DSWD\_TLC39**

```

{
  bool bEnabled;
}

```

- **bEnabled**

whether TLC39 is enabled or disabled

### 1.2.60 CONFIG\_2DSWD\_TRIOPTIC

2D바코드 스캐너에서 지원되는 Trioptic Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_TRIOPTIC](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether Trioptic is enabled or disabled

### 1.2.61 CONFIG\_2DSWD\_UPCA

2D바코드 스캐너에서 지원되는 UPC-A Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_UPCA](#)

```
{
    bool bEnabled;
    bool bAddenda2Digit;
    bool bAddenda5Digit;
    bool bAddendaReq;
    bool bAddendaSeparator;
    bool bXmitCheckDigit;
    bool bXmitNumSys;
}
```

- **bEnabled**

whether UPC-A is enabled or disabled

- **bAddenda2Digit**

BOOL variable that determines if the engine will look for a 2 digit addenda at the end of the UPC bar code. If TRUE, and an addenda is present, the engine adds the two digit addenda data to the end of the message. If FALSE, the engine ignores addenda data. the original or the AIM specification.

- **bAddenda5Digit**

BOOL variable that determines if the engine will look for a 5 digit addenda at the end of the UPC bar code. If TRUE, and an addenda is present, the engine adds the five digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddendaReq**

BOOL variable that determines if the engine will decode only UPC bar codes that have a 2 or 5 digit addenda. If TRUE, the engine decodes only UPC symbols with an addenda. If FALSE, the engine decodes all enabled UPC symbols.

- **bAddendaSeparator**

BOOL variable that determines if there is a space character between the data from the bar code and the data from the addenda. If TRUE, there is a space. If FALSE, there is no space.

- **bXmitCheckDigit**

BOOL variable that determines if the engine will return the check digit as part of the data string after a successful decode. If TRUE, the engine returns the check digit. If FALSE, the check digit is not returned.

- **bXmitNumSys**

BOOL variable that determines if the engine will return the numeric system digit of the UPC label. If TRUE, the engine returns the number system digit. If FALSE, the number system digit is not returned.

## 1.2.62 CONFIG\_2DSWD\_UPCE

2D바코드 스캐너에서 지원되는 UPC-E Symbology의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_UPCE](#)

```
{
    bool bAddenda2Digit;
    bool bAddenda5Digit;
    bool bAddendaReq;
    bool bAddendaSeparator;
    bool bE0Enabled;
    bool bE1Enabled;
    bool bExpandVersionE;
    bool bXmitCheckDigit;
    bool bXmitNumSys;
}
```

- **bAddenda2Digit**

BOOL variable that determines if the engine will look for a 2 digit addenda at the end of the UPC bar code. If TRUE, and an addenda is present, the engine adds the two digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddenda5Digit**

BOOL variable that determines if the engine will look for a 5 digit addenda at the end of the UPC bar code. If TRUE, and an addenda is present, the engine adds the five digit addenda data to the end of the message. If FALSE, the engine ignores addenda data.

- **bAddendaReq**

BOOL variable that determines if the engine will decode only UPC bar codes that have a 2 or 5 digit addenda. If TRUE, the engine decodes only UPC symbols with an addenda. If FALSE, the engine decodes all enabled UPC symbols.

- **bAddendaSeparator**

BOOL variable that determines if there is a space character between the data from the bar code and the data from the addenda. If TRUE, there is a space. If FALSE, there is no space.

- **bXmitCheckDigit**

BOOL variable that determines if the engine will return the check digit as part of the data string after a successful decode. If TRUE, the engine returns the check digit. If FALSE, the check digit is not returned.

- **bXmitNumSys**

BOOL variable that determines if the engine will return the numeric system digit of the UPC label. If TRUE, the engine returns the number system digit. If FALSE, the number system digit is not returned.

### 1.2.63 CONFIG\_2DSWD\_USPS4CB

2D바코드 스캐너에서 지원되는 USPS 4-State Customer Barcode의 설정 정보를 저장하는 구조체

Public struct [CONFIG\\_2DSWD\\_USPS4CB](#)

```
{
    bool bEnabled;
}
```

- **bEnabled**

whether USPS 4-State Customer Barcode is enabled or disabled

#### 1.2.64 CONFIG\_2DSWD\_IDTAG

2D바코드 스캐너에서 지원되는 IDTAG Symbology의 설정 정보를 저장하는 구조체

Public struct **CONFIG\_2DSWD\_IDTAG**

```
{  
    bool bEnabled;  
}
```

- **bEnabled**

whether IDTAG is enabled or disabled

### 1.3 Delegates

#### 1.3.1 BARCODECALLBACK

바코드 인식을 성공했을 때 호출될 delegate 함수.

응용프로그램에서 바코드 데이터를 처리하기 위해서는 SetCallback(BARCODECALLBACK pFunc) 함수를 사용해서 delegate 함수를 만드시 등록 해야 한다.

Public delegate void **BARCODECALLBACK**();



## 1.4 Methods

### 1.4.1 Open

Barcode Scanner Device에 전원을 인가하고 스캐너 디바이스를 초기화 한다. 필요한 System resource를 할당 받는다.

```
BARCODE_RESULT Open();
```

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### 1.4.2 Close

Barcode Scanner Device의 전원을 제거하고 할당 받은 Resource를 해제 한다.

```
BARCODE_RESULT Close();
```

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### 1.4.3 IsOpened

Barcode Scanner Device가 이미 열려 있는지 확인한다.

```
BOOL IsOpened()
```

#### Parameters

*None*

#### Return Values

Barcode Scanner Device의 open 상태.

True:Open, False:Closed

### 1.4.4 SetCallback

Barcode가 인식 완료 되었을 때 호출되는 delegate 함수를 등록한다.

```
BARCODE_RESULT SetCallback (  
    BARCODECALLBACK pFunc,  
);
```

**Parameters***pFunc*

BARCODECALLBACK delegate 함수

**Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

**1.4.5 GetVersionInfo**

바코드 스캐너의 버전을 읽는다.

현재는 사용하지 않음.

```
BARCODE_RESULT GetVersionInfo ( );
```

**Parameters***None***Return Values**

이 함수는 항상 BARCODE\_RESULT\_UNSUPPORTED를 반환한다.

**1.4.6 Start**

비동기적으로 Barcode 읽기를 시작한다.

```
BARCODE_RESULT Start();
```

**Parameters***None***Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

**Notes**

성공적으로 수행되면 Barcode 읽기를 성공하거나, Stop 함수를 호출하기 전까지 Barcode 읽기를 시도한다. Barcode 읽기가 성공적으로 완료 되면 BARCODECALLBACK delegate를 실행 한다.

#### 1.4.7 StartScanRawData

비동기적으로 Barcode 읽기를 시작한다. 읽기의 결과는 string이 아닌 byte 배열이다.

```
BARCODE_RESULT StartScanRawData();
```

##### Parameters

*None*

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

성공적으로 수행되면 Barcode 읽기를 성공하거나, Stop 함수를 호출하기 전까지 Barcode 읽기를 시도한다. Barcode 읽기가 성공적으로 완료 되면 BARCODECALLBACK delegate를 실행 한다.

2D Barcode Scanner에만 해당 됨.

#### 1.4.8 Stop

Barcode 읽기를 중단한다.

```
BARCODE_RESULT Stop();
```

##### Parameters

*None*

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### 1.4.9 StopScanRawData

Barcode 읽기를 중단한다.

```
BARCODE_RESULT StopScanRawData();
```

##### Parameters

*None*

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### 1.4.10 GetScannedData

인식된 바코드 정보를 읽어 온다.

```

BARCODE_RESULT GetScannedData (
    ref String BarcodeValue,
    ref String BarcodeTypeName,
    ref String BarcodeTypeId,
);

```

#### Parameters

*BarcodeValue*

Barcode 값.

*BarcodeTypeName*

Barcode Type Name

*BarcodeTypeId*

Barcode Type ID

#### Return Values

성공적으로 수행되면 **BARCODE\_RESULT\_SUCCESS**를 반환한다.

#### Notes

이 함수는 반드시 **BARCODECALLBACK** delegate 함수 내에서 호출해야 한다.

### 1.4.11 GetBarcodeRawData

인식된 바코드의 raw 데이터를 읽어 온다.

```

BARCODE_RESULT GetBarcodeRawData (
    ref byte[] BarcodeValue,
    ref int Length,
    ref String BarcodeTypeName,
    ref String BarcodeTypeId,
);

```

#### Parameters

*BarcodeValue*

바코드의 raw 데이터가 저장될 byte 배열

*Length*

저장된 raw 데이터의 길이

*BarcodeTypeName*

Barcode Type Name

*BarcodeTypeId*

Barcode Type ID

### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

### Notes

이 함수는 반드시 `BARCODECALLBACK` delegate 함수 내에서 호출해야 한다.

#### 1.4.12 InitCapture

2D Barcode Scanner를 사용해서 이미지 캡처를 할 때, 필요한 System resource를 할당 받고 관련 parameter를 초기화 한다.

`BARCODE_RESULT` InitCapture ();

### Parameters

*None*

### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

### Notes

2D Barcode Scanner에만 해당 됨.

#### 1.4.13 DeinitCapture

이미지 캡처를 위해 할당 받은 System resource를 해제 한다.

`BARCODE_RESULT` DeinitCapture ();

### Parameters

*None*

### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

### Notes

2D Barcode Scanner에만 해당 됨.

#### 1.4.14 SetPreviewHwnd

2D Barcode Scanner를 사용해서 이미지 캡처를 할 때, 미리 보기 화면의 windows handle 을 설정한다.

`BARCODE_RESULT` SetPreviewHwnd (  
    IntPtr pHandle,  
);

**Parameters***pHandle*

미리 보기 화면의 windows handle.

**Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

**Notes**

2D Barcode Scanner에만 해당 됨.

**1.4.15 StartPreview**

2D Barcode Scanner로부터 수신된 Image Data를 미리 보기 화면에 drawing 한다.

BARCODE\_RESULT StartPreview ();

**Parameters***None***Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

**Notes**

2D Barcode Scanner에만 해당 됨.

**1.4.16 StopPreview**

미리 보기를 중단한다.

BARCODE\_RESULT StopPreview ();

**Parameters***None***Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

**Notes**

2D Barcode Scanner에만 해당 됨.

**1.4.17 DoCapture**

2D Barcode Scanner로부터 입력되는 Image data를 파일로 저장한다.

BARCODE\_RESULT DoCapture (

```

    ref BARCODECAPTUREPARAMS CaputreParams
);

```

#### Parameters

*CaptureParams*

저장될 이미지의 정보가 포함된 구조체

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

2D Barcode Scanner에만 해당 됨.

### 1.4.18 GetEnableStateAll

모든 symbology에 대한 Eanble/Disable 상태를 반환한다.

```

BARCODE_RESULT GetEnableStateAll (
    ref bool[] bSymbolTable
);

```

#### Parameters

*bSymbolTable*

Enable/Disable 상태가 저장될 bool 배열

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

pSymbolTable의 크기는 반드시 NUM\_OF\_1D\_SYMBOLOGIES 또는 NUM\_OF\_2DSWD\_SYMBOLOGIES로 설정되어야 하며, OCR의 경우에는 Symbology의 특성상 상태 확인이 어려우므로, OCR 설정 값을 직접 읽어서(GetSymbologyConfig) 확인 할 것을 권장한다.

### 1.4.19 EnableSymbologiesAll

모든 symbology에 대한 Eanble/Disable 상태를 설정한다.

```

BARCODE_RESULT GetEnableStateAll (
    bool bEnable
);

```

#### Parameters

*bEnable*

모든 symbology에 적용할 Enable/Disable 상태

True: 모든 symbology가 Enable 됨

False: 모든 symbology가 Disable 됨

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

#### 1.4.20 EnableSymbologies

특정 Symbology를 enable 한다.

```
BARCODE_RESULT EnableSymbologies (
    bool[] bSymbolTable
);
```

#### Parameters

*bSymbolTable*

enable할 Symbology의 정보가 저장된 bool 배열

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

bSymbolTable의 크기는 반드시 NUM\_OF\_1D\_SYMBOLOGIES 또는 NUM\_OF\_2DSWD\_SYMBOLOGIES로 설정되어야 하며, SYMBOLOGIES\_1D 또는 SYMBOLOGIES\_2DSWD에 정의 되어 있는 값을 사용하여 enable할 Sybology를 true로 설정한다.

#### 1.4.21 DisableSymbologies

특정 Symbology를 Disable 한다.

```
BARCODE_RESULT DisableSymbologies (
    bool[] bSymbolTable
);
```

#### Parameters

*bSymbolTable*

disable할 Symbology의 정보가 저장된 bool 배열

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes



bSymbolTable의 크기는 반드시 NUM\_OF\_1D\_SYMBOLOGIES 또는 NUM\_OF\_2DSWD\_SYMBOLOGIES로 설정되어야 하며, SYMBOLOGIES\_1D 또는 SYMBOLOGIES\_2DSWD에 정의 되어 있는 값을 사용하여 disable할 Sybology를 true로 설정한다.

#### 1.4.22 GetSymbologyConfig

특정 Symbology의 설정값을 스캐너로부터 읽는다.

```
BARCODE_RESULT GetSymbologyConfig (
    ref SYMBOL_CONFIG SymbolConfig
);
```

##### Parameters

*SymbolConfig*

스캐너로부터 읽어온 Symbology의 설정값이 저장될 구조체

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

1.2.3 SYMBOL\_CONFIG 참조

#### 1.4.23 SetSymbologyConfig

특정 Symbology의 설정값을 스캐너에 저장한다.

```
BARCODE_RESULT SetSymbologyConfig (
    SYMBOL_CONFIG SymbolConfig
);
```

##### Parameters

*SymbolConfig*

스캐너에 저장할 Symbology의 설정 값이 저장되어있는 구조체

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

1.2.3 SYMBOL\_CONFIG 참조

#### 1.4.24 GetGeneralConfig

1D 바코드 스캐너의 security level, scanning angle등을 설정 값을 읽는다.

```
BARCODE_RESULT GetGeneralConfig (
    ref GENERAL_CONFIG GeneralConfig
);
```

#### Parameters

*GeneralConfig*

스캐너로부터 읽어온 설정 값이 저장될 구조체

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

1D Barcode Scanner에만 해당 됨.

1.2.2 GENERAL\_CONFIG 참조.

#### 1.4.25 SetGeneralConfig

1D 바코드 스캐너의 security level, scanning angle등의 값을 설정한다.

```
BARCODE_RESULT SetGeneralConfig (
    GENERAL_CONFIG GeneralConfig
);
```

#### Parameters

*GeneralConfig*

스캐너에 저장할 설정 값이 저장되어있는 구조체

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

1D Barcode Scanner에만 해당 됨.

1.2.2 GENERAL\_CONFIG 참조.

#### 1.4.26 SetDefaultSymbol

모든 Symbology의 설정 값을 기본값으로 되돌린다.

```
BARCODE_RESULT SetDefaultSymbol ();
```

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

#### 1.4.27 SetScanningLightMode

2D 바코드 스캐너가 스캔을 진행하는 동안의 LED light(aimers, illumination) 설정을 한다.

```
BARCODE_RESULT SetScanningLightMode (
    int nMode
);
```

#### Parameters

*nMode*

- 0 : Neither aimers nor illumination
- 1 : Illumination only
- 2 : Aimers only
- 3 : Both aimers and illumination

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

#### 1.4.28 SetLeaveLightsOn

2D 바코드 스캐너가 스캔을 진행하는 동안의 LED light(aimers, illumination)의 점멸 여부r 값을 설정한다.

```
BARCODE_RESULT SetLeaveLightsOn (
    bool bEnable
);
```

#### Parameters

*bEnable*

- true : 스캔하는 동안 LED가 점멸하지 않고 항상 On 상태임.
- false : 스캔하는 동안 LED가 점멸함.

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

#### 1.4.29 GetLeaveLightsOn

2D 바코드 스캐너가 스캔을 진행하는 동안의 LED light(aimers, illumination)의 점멸 여부를 값을 읽는다.

```
BARCODE_RESULT GetLeaveLightsOn (  
    ref bool bEnable  
);
```

##### Parameters

*bEnable*

true : 스캔하는 동안 LED가 점멸하지 않고 항상 On 상태임.

false : 스캔하는 동안 LED가 점멸함.

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

## 1.5 Properties

#### 1.5.1 IlluminationPowerLevel

2D Scanner 엔진의 Illumination Power 값을 읽거나 설정한다.

```
byte IlluminationPowerLevel{ get; set;}
```

##### Value

Power level (min:0 ~ max:255)

## 2 C/C++ API Reference

### 2.1 Enumerations

#### 2.1.1 BARCODE\_RESULT

함수의 호출 결과를 나타낸다.

- **BARCODE\_RESULT\_ALREADY\_BARCODE\_SELECTED**  
이미 바코드 Device가 선택되어 있음.
- **BARCODE\_RESULT\_ALREADY\_OPENED**  
바코드 Device가 이미 열려 있음.
- **BARCODE\_RESULT\_BARCODE\_INIT\_FAILURE**  
바코드 스캐너 초기화 실패.
- **BARCODE\_RESULT\_CAM\_SELECTED**  
CAM Device를 사용중임.
- **BARCODE\_RESULT\_FAILURE**  
기능 수행 실패.
- **BARCODE\_RESULT\_INVALID\_ARGS**  
유효하지 않은 파라미터.
- **BARCODE\_RESULT\_INVALID\_DEVICE**  
바코드 스캐너가 장착되어 있지 않음.
- **BARCODE\_RESULT\_NOT\_OPENED**  
Open을 호출하지 않고 기능 수행을 시도 함.
- **BARCODE\_RESULT\_OUTOFMEMORY**  
바코드 스캐너 자원 할당 실패.
- **BARCODE\_RESULT\_SUCCESS**  
기능 수행 성공.
- **BARCODE\_RESULT\_UNSUPPORTED**  
지원 되지 않는 명령.

## 2.2 Constants

### 2.2.1 WM\_BARCODE\_ONSCANED

Barcode를 인식 완료 했을 때 Application으로 전달되는 Message.

- #define **WM\_BARCODE\_ONSCANED**      **WM\_USER + 1801**

### 2.2.2 CAP\_IMG\_VGA

2D Barcode Scanner를 이용하여 캡처 할 이미지의 사이즈

640 \* 480

- #define **CAP\_IMG\_VGA**      **1**

### 2.2.3 CAP\_IMG\_QVGA

2D Barcode Scanner를 이용하여 캡처 할 이미지의 사이즈

320 \* 240

- #define **CAP\_IMG\_VGA**      **2**

### 2.2.4 CAP\_IMG\_PREVIEW

2D Barcode Scanner를 이용하여 캡처 할 이미지의 사이즈

240 \* 180

- #define **CAP\_IMG\_VGA**      **3**

### 2.2.5 IMG\_FORMAT\_JPEG

2D Barcode Scanner를 이용하여 캡처 할 이미지의 포맷

Jpeg format

- #define **IMG\_FORMAT\_JPEG**      **1**

### 2.2.6 IMG\_FORMAT\_BMP

2D Barcode Scanner를 이용하여 캡처 할 이미지의 포맷

BMP format

- #define **IMG\_FORMAT\_BMP**      **2**

## 2.3 Structures

### 2.3.1 BARCODECAPTUREPARAMS

2D Barcode Scanner를 이용해서 이미지를 캡처 할 때, 이미지 정보를 저장하는 구조체.

```
typedef struct
{
    char ImgFormat
    char ImgSize;
    TCHAR szFilePath[MAX_PATH];
} BARCODECAPTUREPARAMS;
```

- **ImgFormat**

이미지 포맷.

- **ImgSize**

이미지 사이즈.

- **szFilePath**

파일명을 포함한 이미지 저장 경로.

### 2.3.2 BARCODE\_GENERAL\_CONFIG

1D 바코드 스캐너의 security level, scanning angle 상태를 읽거나 설정할 때 사용하는 구조체

```
typedef struct
{
    BOOL bBiDirectionalRedundancy;
    int nAngle;
    int nLinearCodeSecurityLevel
} BARCODE_GENERAL_CONFIG;
```

- **bBiDirectionalRedundancy**

Returns whether a bar code must be successfully scanned in both direction(forward and reverse) before being decoded.

- **nAngle**

Scanning angle. 5=Narrow, 6=Wide

- **nLinearCodeSecurityLevel**

Four levels(1~4) of decode security for linear code types are supported. Select a

higher security level for low levels of bar code quality. As security levels increase, the decoder's aggressiveness decrease.

### 2.3.3 BARCODE\_SYMBOL\_CONFIG

Symbology 설정을 위하여 설정 정보를 담는 구조체.

```
typedef struct
{
    Void * pConfigData;
    Int Symbol;
} BARCODE_SYMBOL_CONFIG;
```

**- pConfigData**

설정을 변경할 Symbology의 구조체의 포인터

**- Symbol**

설정을 변경할 Symbology의 ID

### 2.3.4 Symbol Configuration Structures

..NET API reference의 내용과 동일하므로 생략.

..NET API reference의 1.2.4 부터 1.2.64 까지의 항목을 참조한다.

## 2.4 Callback function definition

### 2.4.1 BARCODECALLBACK

바코드 인식을 성공 했을 때 호출되는 callback 함수.

응용프로그램에서 callback 함수의 실행을 통해서 데이터를 처리하기 위해서는 BarcodeSetCallback() 함수를 사용해서 callback 함수 설정을 해야 한다.

**- typedef void (CALLBACK\* BARCODECALLBACK)();**



## 2.5 Methods

### 2.5.1 BarcodeOpen

Barcode Scanner Device에 전원을 인가하고 스캐너 디바이스를 초기화 한다. 필요한 System resource를 할당 받는다.

**BARCODE\_RESULT** BarcodeOpen();

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### 2.5.2 BarcodeClose

Barcode Scanner Device의 전원을 제거하고 할당 받은 Resource를 해제 한다.

**BARCODE\_RESULT** BarcodeClose();

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### 2.5.3 BarcodeIsOpened

Barcode Scanner Device가 이미 열려 있는지 확인한다.

**BOOL** BarcodeIsOpened()

#### Parameters

*None*

#### Return Values

Barcode Scanner Device의 open 상태.

True:Open, False:Closed

#### 2.5.4 BarcodeSetCallback

Barcode가 인식 완료 되었을 때 호출되는 callback 함수를 등록한다.

```
BARCODE_RESULT BarcodeSetCallback (  
    BARCODECALLBACK pFunc,  
);
```

##### Parameters

*pFunc*

BARCODECALLBACK callback 함수

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### 2.5.5 BarcodeGetVersionInfo

비동기적으로 Barcode 읽기를 시작한다.

```
BARCODE_RESULT BarcodeVersionInfo();
```

##### Parameters

*None*

##### Return Values

이 함수는 항상 BARCODE\_RESULT\_UNSUPPORTED를 반환한다.

#### 2.5.6 BarcodeStart

비동기적으로 Barcode 읽기를 시작한다.

```
BARCODE_RESULT BarcodeStart();
```

##### Parameters

*None*

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

성공적으로 수행되면 Barcode 읽기를 성공하거나, Stop 함수를 호출하기 전까지 Barcode 읽기를 시도한다. Barcode 읽기가 성공적으로 완료 되면 BARCODECALLBACK delegate를 실행 한다.

### 2.5.7 BarcodeStartScanRawData

비동기적으로 Barcode 읽기를 시작한다. 읽기의 결과는 string이 아닌 byte 배열이다.

```
BARCODE_RESULT BarcodeStartScanRawData();
```

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

#### Notes

성공적으로 수행되면 Barcode 읽기를 성공하거나, Stop 함수를 호출하기 전까지 Barcode 읽기를 시도한다.

2D Barcode Scanner에만 해당 됨.

### 2.5.8 BarcodeStop

Barcode 읽기를 중단한다.

```
BARCODE_RESULT BarcodeStop();
```

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### 2.5.9 BarcodeStopScanRawData

Barcode 읽기를 중단한다.

```
BARCODE_RESULT BarcodeStopScanRawData();
```

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### 2.5.10 BarcodeGetScannedData

인식된 바코드 정보를 읽어 온다.

```

BARCODE_RESULT BarcodeGetScannedData (
    LPWSTR BarcodeValue,
    LPWSTR BarcodeTypeName,
    LPWSTR BarcodeTypeId,
);

```

#### Parameters

*BarcodeValue*

Barcode 값.

*BarcodeTypeName*

Barcode Type Name

*BarcodeTypeId*

Barcode Type ID

#### Return Values

성공적으로 수행되면 **BARCODE\_RESULT\_SUCCESS**를 반환한다.

#### Notes

이 함수는 반드시 **WM\_BARCODE\_ONSCANED** 메시지를 받았을 때 또는 **Callback** 함수 내에서 호출해야 한다.

### 2.5.11 BarcodeGetBarcodeRawData

인식된 바코드의 raw 데이터를 읽어 온다.

```

BARCODE_RESULT BarcodeGetBarcodeRawData (
    BYTE * BarcodeValue,
    WORD * pLength,
    LPWSTR BarcodeTypeName,
    LPWSTR BarcodeTypeId,
);

```

#### Parameters

*BarcodeValue*

바코드의 raw 데이터가 저장될 byte 배열의 포인터

*pLength*

raw 데이터의 길이가 저장될 변수의 포인터

*BarcodeTypeName*

Barcode Type Name

*BarcodeTypeId*

Barcode Type ID

### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

### Notes

이 함수는 반드시 `WM_BARCODE_ONSCANED` 메시지를 받았을 때 또는 `Callback` 함수 내에서 호출해야 한다.

#### 2.5.12 BarcodeInitCapture

2D Barcode Scanner를 사용해서 이미지 캡처를 할 때, 필요한 System resource를 할당 받고 관련 parameter를 초기화 한다.

`BARCODE_RESULT` BarcodeInitCapture ();

### Parameters

*None*

### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

### Notes

2D Barcode Scanner에만 해당 됨.

#### 2.5.13 BarcodeDeinitCapture

이미지 캡처를 위해 할당 받은 System resource를 해제 한다.

`BARCODE_RESULT` BarcodeDeinitCapture ();

### Parameters

*None*

### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

### Notes

2D Barcode Scanner에만 해당 됨.

#### 2.5.14 BarcodeSetPreviewHwnd

2D Barcode Scanner를 사용해서 이미지 캡처를 할 때, 미리 보기 화면의 windows handle 을 설정한다.

`BARCODE_RESULT` BarcodeSetPreviewHwnd (  
`HWND` hWnd,

);

#### Parameters

*hWnd*

미리 보기 화면의 windows handle.

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

2D Barcode Scanner에만 해당 됨.

#### 2.5.15 BarcodeStartPreview

2D Barcode Scanner로부터 수신된 Image Data를 미리 보기 화면에 drawing 한다.

`BARCODE_RESULT` BarcodeStartPreview ();

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

2D Barcode Scanner에만 해당 됨.

#### 2.5.16 BarcodeStopPreview

미리 보기를 중단한다.

`BARCODE_RESULT` BarcodeStopPreview ();

#### Parameters

*None*

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

2D Barcode Scanner에만 해당 됨.

#### 2.5.17 BarcodeDoCapture

2D Barcode Scanner로부터 입력되는 Image data를 파일로 저장한다.

```

BARCODE_RESULT BarcodeDoCapture (
    BARCODECAPTUREPARAMS* pCaputreParams
);

```

#### Parameters

*pCaputreParams*

저장될 이미지의 정보가 포함된 구조체

#### Return Values

성공적으로 수행되면 **BARCODE\_RESULT\_SUCCESS**를 반환한다.

#### Notes

2D Barcode Scanner에만 해당 됨.

### 2.5.18 BarcodeGetEnableStateAll

모든 symbology에 대한 Eanble/Disable 상태를 반환한다.

```

BARCODE_RESULT BarcodeGetEnableStateAll (
    BOOL * bSymbolTable
);

```

#### Parameters

*pSymbolTable*

Enable/Disable 상태가 저장될 **BOOL** 배열의 포인터

#### Return Values

성공적으로 수행되면 **BARCODE\_RESULT\_SUCCESS**를 반환한다.

#### Notes

*pSymbolTable*의 크기는 반드시 **NUM\_OF\_1D\_SYMBOLOGIES** 또는 **NUM\_OF\_2DSWD\_SYMBOLOGIES**로 설정되어야 하며, OCR의 경우에는 Symbology의 특성상 상태 확인이 어려우므로, OCR 설정 값을 직접 읽어서(**GetSymbologyConfig**) 확인 할 것을 권장한다.

### 2.5.19 BarcodeEnableSymbologiesAll

모든 symbology에 대한 Eanble/Disable 상태를 설정한다.

```

BARCODE_RESULT BarcodeGetEnableStateAll (
    BOOL bEnable
);

```

#### Parameters

### *bEnable*

모든 symbology에 적용할 Enable/Disable 상태

True: 모든 symbology가 Enable 됨

False: 모든 symbology가 Disable 됨

### **Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### **Notes**

#### **2.5.20 BarcodeEnableSymbologies**

특정 Symbology를 enable 한다.

```
BARCODE_RESULT BarcodeEnableSymbologies (
    BOOL * pSymbolTable
);
```

### **Parameters**

*pSymbolTable*

enable할 Symbology의 정보가 저장된 BOOL 배열의 포인터

### **Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### **Notes**

bSymbolTable의 크기는 반드시 NUM\_OF\_1D\_SYMBOLOGIES 또는 NUM\_OF\_2DSWD\_SYMBOLOGIES로 설정되어야 하며, SYMBOLOGIES\_1D 또는 SYMBOLOGIES\_2DSWD에 정의 되어 있는 값을 사용하여 enable할 Sybology를 true로 설정한다.

#### **2.5.21 BarcodeDisableSymbologies**

특정 Symbology를 Disable 한다.

```
BARCODE_RESULT BarcodeDisableSymbologies (
    BOOL * pSymbolTable
);
```

### **Parameters**

*pSymbolTable*

disable할 Symbology의 정보가 저장된 BOOL 배열의 포인터

### **Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.



### Notes

bSymbolTable의 크기는 반드시 NUM\_OF\_1D\_SYMBOLOGIES 또는 NUM\_OF\_2DSWD\_SYMBOLOGIES로 설정되어야 하며, SYMBOLOGIES\_1D 또는 SYMBOLOGIES\_2DSWD에 정의 되어 있는 값을 사용하여 disable할 Sybology를 true로 설정한다.

#### 2.5.22 BarcodeGetSymbologyConfig

특정 Symbology의 설정값을 스캐너로부터 읽는다.

```
BARCODE_RESULT BarcodeGetSymbologyConfig (
    BARCODE_SYMBOL_CONFIG * pSymbolConfig
);
```

### Parameters

*pSymbolConfig*

스캐너로부터 읽어온 Symbology의 설정 값이 저장될 구조체

### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### Notes

2.3.3 BARCODE\_SYMBOL\_CONFIG 참조

#### 2.5.23 BarcodeSetSymbologyConfig

특정 Symbology의 설정값을 스캐너에 저장한다.

```
BARCODE_RESULT BarcodeSetSymbologyConfig (
    BARCODE_SYMBOL_CONFIG * pSymbolConfig
);
```

### Parameters

*pSymbolConfig*

스캐너에 저장할 Symbology의 설정 값이 저장되어있는 구조체

### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### Notes

2.3.3 BARCODE\_SYMBOL\_CONFIG 참조

#### 2.5.24 BarcodeGetGeneralConfig

1D 바코드 스캐너의 security level, scanning angle등을 설정 값을 읽는다.

```
BARCODE_RESULT BarcodeGetGeneralConfig (
    BARCODE_GENERAL_CONFIG * pGeneralConfig
);
```

##### Parameters

*pGeneralConfig*

스캐너로부터 읽어온 설정 값이 저장될 구조체

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

1D Barcode Scanner에만 해당 됨.

2.3.2 GENERAL\_CONFIG 참조.

#### 2.5.25 BarcodeSetGeneralConfig

1D 바코드 스캐너의 security level, scanning angle등의 값을 설정한다.

```
BARCODE_RESULT BarcodeSetGeneralConfig (
    BARCODE_GENERAL_CONFIG * pGeneralConfig
);
```

##### Parameters

*pGeneralConfig*

스캐너에 저장할 설정 값이 저장되어있는 구조체

##### Return Values

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

##### Notes

1D Barcode Scanner에만 해당 됨.

2.3.2 GENERAL\_CONFIG 참조.

#### 2.5.26 BarcodeSetDefaultSymbol

모든 Symbology의 설정 값을 기본값으로 되돌린다.

```
BARCODE_RESULT BarcodeSetDefaultSymbol ( );
```

##### Parameters

*None*

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

### 2.5.27 BarcodeSetScanningLightMode

2D 바코드 스캐너가 스캔을 진행하는 동안의 LED light(aimers, illumination) 설정을 한다.

```
BARCODE_RESULT BarcodeSetScanningLightMode (
    int nMode
);
```

#### Parameters

*nMode*

- 0 : Neither aimers nor illumination
- 1 : Illumination only
- 2 : Aimers only
- 3 : Both aimers and illumination

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

### 2.5.28 BarcodeSetLeaveLightsOn

2D 바코드 스캐너가 스캔을 진행하는 동안의 LED light(aimers, illumination)의 점멸 여부r 값을 설정한다.

```
BARCODE_RESULT BarcodeSetLeaveLightsOn (
    BOOL bEnable
);
```

#### Parameters

*bEnable*

- true : 스캔하는 동안 LED가 점멸하지 않고 항상 On 상태임.
- false : 스캔하는 동안 LED가 점멸함.

#### Return Values

성공적으로 수행되면 `BARCODE_RESULT_SUCCESS`를 반환한다.

#### Notes

### 2.5.29 BarcodeGetLeaveLightsOn

2D 바코드 스캐너가 스캔을 진행하는 동안의 LED light(aimers, illumination)의 점멸 여부를 값을 읽는다.

```

BARCODE_RESULT BarcodeGetLeaveLightsOn (
    BOOL * pbEnable
);

```

#### Parameters

*pbEnable*

TRUE : 스캔하는 동안 LED가 점멸하지 않고 항상 On 상태임.

FALSE : 스캔하는 동안 LED가 점멸함.

#### Return Values

성공적으로 수행되면 **BARCODE\_RESULT\_SUCCESS**를 반환한다.

#### Notes

### 2.5.30 BarcodeGetIlluminationPowerLevel

2D 바코드 스캐너의 Illumination power level을 읽는다.

```

BARCODE_RESULT BarcodeGetIlluminationPowerLevel (
    BYTE* pPwrLevel
);

```

#### Parameters

*pPwrLevel*

illumination power level

#### Return Values

성공적으로 수행되면 **BARCODE\_RESULT\_SUCCESS**를 반환한다.

#### Notes

스캐너가 Motorola SE4500일 경우만 지원된다.

### 2.5.31 BarcodeSetIlluminationPowerLevel

2D 바코드 스캐너의 Illumination power level을 설정한다.

```

BARCODE_RESULT BarcodeSetIlluminationPowerLevel (
    BYTE PwrLevel
);

```

);

### **Parameters**

*PwrLevel*

illumination power level

### **Return Values**

성공적으로 수행되면 BARCODE\_RESULT\_SUCCESS를 반환한다.

### **Notes**

스캐너가 Motorola SE4500일 경우만 지원된다.